The local query performed in step **430** can take any appropriate form and will depend on the embodiment and the format of the received query. For example, if the received query is an SQL query and the metadata is stored in a database, then the received SQL query or some modification thereof may be used to query the local data. If the received query is a Boolean search and the metadata is stored in a flat file, then a text search based on the Boolean query may be performed. Once matching metadata is found, then the associated CAS content is retrieved and made ready to send to the requester.

In steps **440** and **450**, the CAS server may await results from the other CAS servers. All of the received results are combined and forwarded in step **460**. For example, if a CAS server storing DICOM images requested more results for a particular patient name and patient ID, then the DICOM CAS server may send the request to multiple CAS servers in order to get any results for that patent stored on any of the other servers. The results may be checked for validity via a checksum or other assurance mechanism built into the system, or any other appropriate CAS procedures. Corrupted or otherwise invalid files may be excluded from or flagged in the CAS or in the result set sent to the requester.

Given the nature of the redundancy and distribution that may be possible with embodiments herein, multiple copies of the same CAS data may be received from multiple CAS servers. In such cases, any duplicate files may be excluded from or flagged in the result set sent to the requester. Additionally, in some embodiments, if the remote CAS server does not send a response within a particular timeout period, the CAS server may no longer await results and may send out any received results in step **450**.

The results sent to the requester in step **460** can take any appropriate form, including a single compressed file, a pointed to one or more accessible pieces of data that comprise the complete data set, or any other appropriate mechanism.

FIG. **5** depicts a block diagram of an exemplary process for retrieving CAS data via an application server. The process depicted in FIG. **5** is similar to that depicted in FIG. **4**, except that the search is initiated first by a user to an application server. The application server then sends the request to multiple CAS servers. For example, a user using a client system **140**A sends a request for data to an application server **130**, the application server **130** receives the request (step **510**), the application server **130** forwards the request to the CAS servers **110**A-C (step **520**), the application server searches locally (step **530**), and receives or times out waiting for results from the CAS servers (steps **540** and **550**). Once all of the results are received and compiled as above, the results are sent to the client system **140**A (step **560**).

The local search in step **530** at the application server may be for CAS data, or it may be for other application data. For example, if the CAS servers **110**A-C store DICOM images and the application server **130** stores patient demographic and billing information, then the local results may be related to the demographics and billing history of the results and the results from the CAS servers **110**A-C may be DICOM images.

As above, the results from the CAS servers (and the application server if it stores replicated data) may be checked not only for integrity (via checksums, e.g.), but also for duplicates. The duplicate results may be flagged or omitted from the search results sent.

The processes and systems described herein may be performed on or encompass various types of hardware, such as computer systems. In some embodiments, the computer systems such as the client system **140**A-C, the application server **130**, and the content addressable storage systems **110**A-C may include a bus or other communication mechanism for communicating information, and a processor coupled with the bus for processing information. The computer systems may have a main memory, such as a random access memory or other dynamic storage device, coupled to the bus. The main memory may be used to store instructions and temporary variables. The computer systems may also include a read-only memory or other static storage device coupled to the bus for storing static information and instructions. The computer systems may also be coupled to a display, such as a CRT or LCD monitor. Input devices may also be coupled to the computer system. These input devices may include a mouse, a trackball, or cursor direction keys. Each computer system may be implemented using one or more physical computers or computer systems or portions thereof. The instructions executed by the computer system may also be read in from a computer-readable medium. The computer-readable medium may be a CD, DVD, optical or magnetic disk, laserdisc, carrier wave, or any other medium that is readable by the computer system. In some embodiments, hardwired circuitry may be used in place of or in combination with software instructions executed by the processor.

As will be apparent, the features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure.

Conditional language used herein, such as, among others, “can,” “could,” “might,” “may,” “e.g.,” and the like, unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or states. Thus, such conditional language is not generally intended to imply that features, elements and/or states are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without author input or prompting, whether these features, elements and/or states are included or are to be performed in any particular embodiment.

Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art.

All of the methods and processes described above may be embodied in, and fully automated via, software code modules executed by one or more general purpose computers or processors, such as those computer systems described above. The code modules may be stored in any type of computer-readable medium or other computer storage device. Some or all of the methods may alternatively be embodied in specialized computer hardware.

It should be emphasized that many variations and modifications may be made to the above-described embodiments, the elements of which are to be understood as being among other acceptable examples. All such modifications and variations are intended to be included herein within the scope of this disclosure and protected by the following claims.